Towards Real-time Insect Motion Capture

Deschanel Li*

The Study, Montreal, Canada

Abstract

It is currently possible to reliably motion-track humans and some animals, but not possible to track insects using standard motion tracking techniques. By programming a virtual prototype rig/skeleton for the insects small scale creatures will be able to be tracked in real time. Possible applications include behavioural research of animals and entertainment industry, e.g., when realistic insect motion simulation is needed and insects cannot be outfitted with sensors like humans for animation in movies or games.

Keywords: motion capture, real-time, OpenCV, OpenGL, frameworks, Kinect, Processing, life sciences

Concepts:•Human-centered computing → Graphics input devices;•Computing methodologies → Motion capture; Tracking; •Applied computing → *Life and medical sciences;*

1 Introduction

In this research project we introduce the necessary steps to enable real-time or post-processing motion tracking techniques for insects and small animals. Possible applications include behavioural recognition research of animals and entertainment industry, e.g., when realistic insect motion simulation is needed and insects cannot be outfitted with sensors like humans for movies (e.g., *Ant Man*) or games. In the latter, depending on the size of target, 3D models may potentially be reconstructed. Research in animal behaviour has also recently picked up interest, such has in the *Machine Vision of Animals and their Behaviour (MVAB) 2015* workshop in Swansea, UK (http://di.ncl.ac.uk/mvab2015).

Background. Motion capture is a process of recording the movement of objects or people. The information acquired from recording will then be able to be represented digitally as well as used to animate virtual models.

There are two main types of motion tracking: optical and non-optical.

There are also many techniques that can be used for motion capture such as passive markers, sequential processing using different cameras, video reconstruction using the data of a video taken beforehand and model initiation that can recognize different body parts.

Problem. It is currently possible to motion-track humans and some animals, but not possible to track insects using standard motion tracking techniques using readily available devices and depth cameras, such as Kinect or Structure sensors since, they were trained and optimized for human skeleton, gestures, and postures and larger bodies and distances. Likewise, many databases exist with human motion data from either wearable sensors or Kinect recordings, but no data for insects and small animals other than standard video recordings. It is also impossible to outfit insects and small animals with sensors using current technology.

Proposed Solution. By programming a virtual rig/skeleton for the insect and training the computer to recognize it, small scale creatures will be able to be tracked in realtime. While we cannot use built-in skeleton tracking of Kinect sensors for this task, we can still use its depth image, even at closer distances, by inverting it with the background. The depth data are also useful for subtracting most of the irrelevant background image. Then skeletonizing the remaining image gives us a thin, one-pixel-wide pattern of the target object or a contour. The resulting features can later be used to construct motion data for insects and small animals. This solution can work not only for depth data that we record in real-time but also for pre-recorded colour videos (e.g., the spider stock video in Figure 1).

2 Methodology

The first few tests were done with people, more specifically hands (see Figure 1). They were readily available at any second and served as good initial models since they did somehow resemble a non-human skeletal insect, with the palm being the body and the fingers being the legs. Then, smaller things were used such as models of people and toy spiders. This resulted in the body of the subject being created as a blob in OpenCV, and then the limbs being fully skeletonized using the Zhang-Suen [Zhang and Suen 1984] transform. This is semi-accurate for insect anatomy also, since the thorax and abdomen make up the body of the insect while the legs are thin and stretched out. However, it was noted that although skeletonization worked well, smaller objects were less likely to be skeletonized or even seen since the Kinect wasn't able to recognize objects that small and would clump the little details together. This was especially true for the lower-resolution Kinect v1, whereas Kinect v2 fared better.

Bringing the subject closer to the Kinect would help, but it would also be out of the visible range of the Kinect. However, depth inversion trick here enabled the necessary tracking when blocking off the view of the background by the object in question enabling to track it.

The color camera or video are also used and can help disambiguate depth data at close ranges, but are light-sensitive. A combination of OpenCV-based background removal and skeletonization allows to track, e.g., a spider in the video recording in Figure 1.

Thus, dual depth and color data methodology works with their own individual limitations, to extract the motion data and skeletonize for further learning and processing.

Tools. The following hardware, development and image processing tools enabled the prototype project execution in real-time:

- Microsoft Kinect v1 and v2 depth cameras.
- Processing 2.2.1 [Fry and Reas 2001–2015] – the IDE was used to program a Java-base sketch that is easy to work with and locate the necessary libraries for Kinect, image, and videop processing.
- OpenCV for Processing [Borenstein 2013] – the Processing version of the popular computer vision library for edge detection and contour analysis.
- OpenKinect for Processing – a Processing wrapper of `libfreenect libfreenect2` allowing to connect one or more Kinect v1 or Kinect v2 sensors simultaneously.
- OCTMARF [Mokhov and Sun 2011–2014] – a Java library for Processing that has Zhang-Suen skeletonization [Zhang and Suen 1984] and feature extraction for depth data.
- While most of the work was done on OS X 10.10.5, due to the portable nature of tools it will work on Windows and Linux.

*dli@thestudy.qc.ca,deschaneli@gmail.com

SIGGRAPH '16, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4371-8/16/07
DOI: http://dx.doi.org/10.1145/2945078.2945115

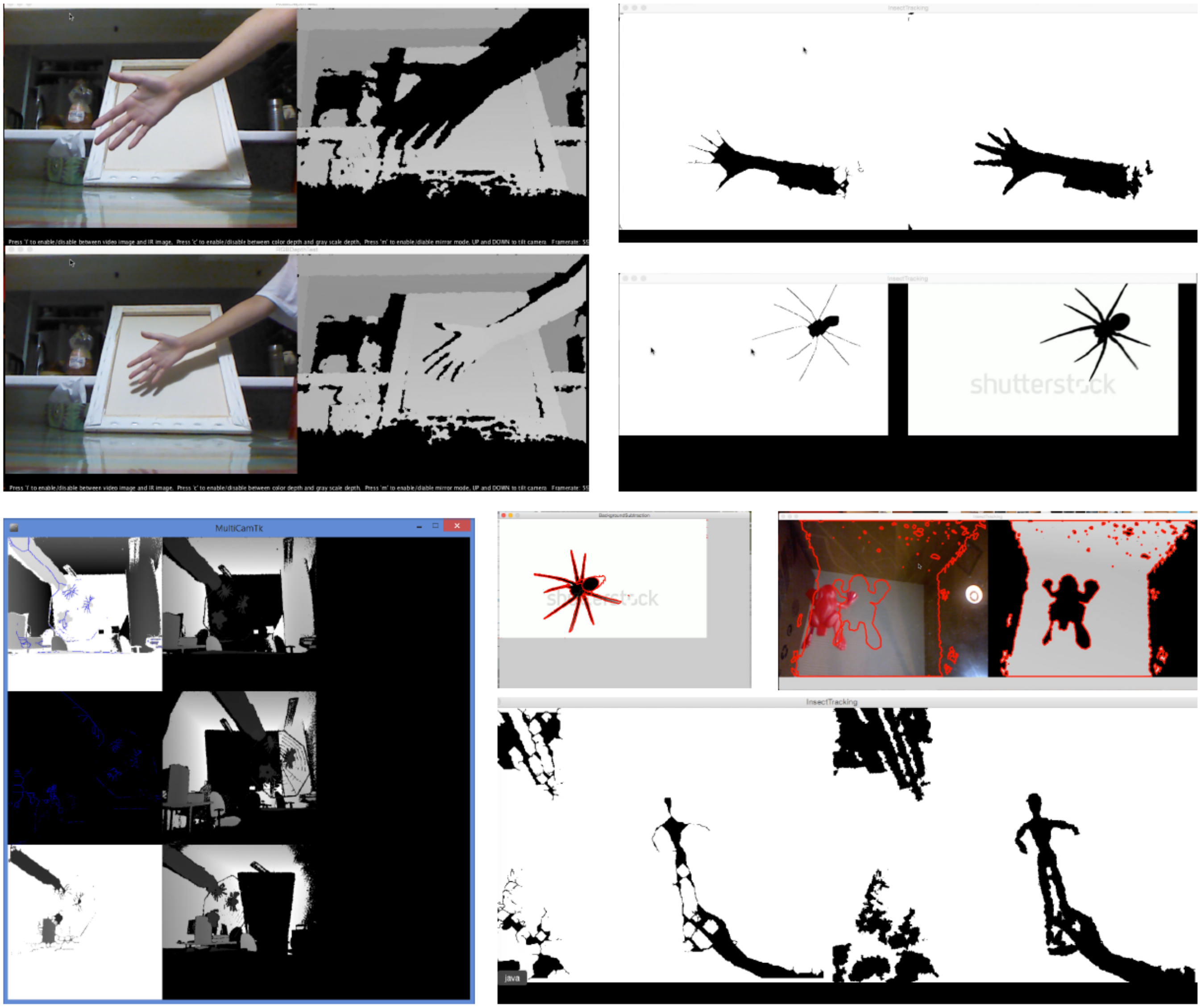

Figure 1: *Prototype Results for Skeletonization and Contouring using both Kinect v1 and Kinect v2*

3 Conclusion

To conclude, the InsectTracker program is able to track and skeletonize large subjects at a medium range easily and smaller objects with Kinect v2 better than Kinect v1. This is also all done in realtime. The original goal was to capture insects, since they were on a small scale, and although this has not been fully achieved yet due to missing motion feature extraction from the skeletonized data. The depth inversion trick as well as near-mode of Kinect help with skeletonizing near-placed objects, but the results need to be more stabilized. Color-based data may also help to combine with the depth data to track better smaller and near insects.

Currently, only data are being captured and converted into something the computer can understand but nothing is done with this data, such as joint feature extraction from the skeleton. For example, as a follow-up, an algorithm can be written to analyze the data collected and distinguish the behavioural traits of a live subject based on their skeletons. The skeleton can also be of use in areas other than science such as animation and/or film.

Future work. Future work includes solving the limitations mentioned earlier and in particular tracking joint points of features extracted from the skeletonized data over time and mapping that data We plan to explore other depth sensors, such as Structure sensor and Kinect v2 more in detail as most of the work was carried out using Kinect v1 sensor. Using multiple depth cameras is another consideration to improve the tracking capabilities (OpenKinect for Processing supports multiple Kinects).

References

BORENSTEIN, G., 2013. OpenCV for Processing. [online], July. https://github.com/atduskgreg/opencv-processing.

FRY, B., AND REAS, C., 2001–2015. Processing – a programming language, development environment, and online community. [online]. http://www.processing.org/.

MOKHOV, S. A., AND SUN, Y., 2011–2014. OCT segmentation survey and summary reviews and a novel 3D segmentation algorithm and a proof of concept implementation. [online]. Online at http://arxiv.org/abs/1204.6725.

ZHANG, T. Y., AND SUEN, C. Y. 1984. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM 27*, 3 (Mar.), 236–239.